A Machine Learning Approach to Identify Phishing Websites: A Comparative Study of Classification Models and Ensemble Learning Techniques

Bhogesh Karthik Gontla [1], Priyanka Gundu [2], Padma Jyothi Uppalapati [3*], Kandula Narasimharao [4] and S Mahaboob Hussain [5]

[1,2,3,4,5]Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, 534201, India

Abstract

Phishing assaults are one of the more prevalent types of cybercrime in the world today. To steal information, users are sent emails and messages. Moreover, websites are used for it. Phishing primarily targets corporate websites, such as those for e-commerce, finance, and governmental organizations. To obtain sensitive user information, attackers impersonate websites, a phenomenon known as phishing. In addition to exploring the use of machine learning algorithms to identify and stop web phishing assaults, this research suggests utilizing machine learning techniques to detect phishing URLs by analyzing various aspects of the URLs. The study includes classification models like Logistic Regression, Random Forest, Decision trees, KNN, Naive Bayes, SVM, and other ensemble learning techniques like Gradient Boosting, XGBoost, Histogram Gradient Boosting, Light Gradient Boosting, and Ada Boost were used to detect phishing websites.

Keywords: Web Phishing, Classification techniques, Ensemble learning, Machine Learning.

Received on 01 May 2023, accepted on 05 June 2023, published on 23 June 2023

doi: 10.4108/eetsis.vi.3300

*Corresponding author. Email: padmajyothi64@gmail.com

1. Introduction

As Internet usage rises and online transactions become more frequent, phishing attempts are a serious security issue that is quickly getting worse. Phishing is the practice of attempting to get sensitive data through electronic contact by impersonating a trustworthy entity to obtain usernames, passwords, credit card numbers, or other private information. Attackers carry out phishing attacks using various methods, including fraudulent emails and websites. Blacklists, which are lists of URLs and Internet Protocol address have been classified as dangerous, are a systematic method for detecting phishing websites. Attackers can easily change the URL to avoid being listed on blacklists using encoding and other techniques [1].

Phishing is a type of cyber-attack that takes consumable data, including credit card numbers and login credentials for accounts. Phishing assaults are becoming more prevalent all around the world [2]. In 2008, 51,401 phishing websites were identified by the Anti-Phishing workgroup. According to a survey by Rivest-Shamir-Adleman (RSA) Security, Inc., phishing attacks cost a global organization $9 billion in 2016 [3]. These figures demonstrate the ineffectiveness of the current phishing attack defences.

Organizations should create a comprehensive plan incorporating technological and non-technical methods to protect against phishing. A few of the technical safeguards include putting a user education program into place, setting up multi-factor authentication, using URL filtering software to block well-known harmful websites, and keeping an eye on network traffic for unusual activities. In addition to constantly evaluating the organization's security posture, non-technical methods include adopting rules and procedures to deal with any potential threats and raising awareness of phishing schemes. By implementing these safeguards, businesses can ensure their users are better protected from phishing scams. Also, organizations should consider investing money into cutting-edge technology like machine

learning or artificial intelligence to spot questionable activity quickly.

Organizations may swiftly identify and mitigate any possible hazards by using these technologies before they become problematic. Lastly, businesses should ensure that their systems are frequently patched and upgraded to guard against the most recent vulnerabilities. Patching systems regularly will assist in lowering the possibility of attackers using known vulnerabilities to access sensitive data [45][46][47][48].

To classify phishing websites, the dataset mainly contains the URL details. The term "Uniform Resource Locator" (URL) refers to the internet address of a web page or other resource. It is a special code that allows users to view a certain web page using a web browser [44]. Several elements comprise a URL, including the protocol, domain name, sub-domain name, path, etc as shown in fig 1.

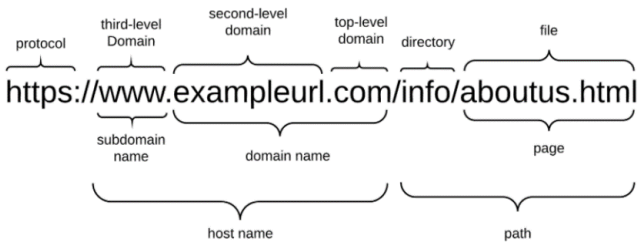

Figure 1. Uniform Resource Locator for a Website

The communication protocol being used to send data over the internet is indicated by the protocol type in a URL (Uniform Resource Locator). The URL starts with the protocol type, widowed by a colon and two forward slashes. These protocols include HTTP, HTTPS, FTP, and SMTP.POP3. The most popular protocol is HTTP, and HTTPS offers far higher security than HTTP, FTP, and SMTP, which are only occasionally used.

Unique names are used to construct domain names. They serve as a website's singular identifier and reside in the URL between the protocol and path. The second-level domain (SLD) and the top-level domain (TLD), respectively.

HTTP://www.exampleurl.com/info/aboutus.html is the URL in question. The domain name "www.exampleurl.com" appears in the URL. The top-level domain is.com.

Second-level domains are "exampleURL". The phisher creates a domain name that is extremely close to an original or legitimate website domain name; the phishing email seems to be from "www.example-url.com" or "www.example_urls.com".

Converting domain names into IP addresses is called a Domain name system (DNS), which websites may then comprehend and use to link user requests to servers and display them on websites.

Additionally, URLs showed to control an attacker entirely. Services like Bitly or TinyURL can be used to conceal the link's true location. The user may find it more challenging to determine where the link will take them.

Following their name or any port number, a path that defines the file's position "aboutus.html" in the directory on the server hosting "www.exampleulr.com" appears. The path may also comprise several directory names or a file name on the server. The path is a crucial part of the URL because it enables the web server to give the client requesting access to a particular resource [4][5].

Using a valid domain name and adding a bogus route to the URL, such as "/login.php," the attacker tricks users into providing their credentials on what appears to be a login or update page. Still, the attackers are accessing the victims' information.

The rest of the paper is organized as follows. Section 2 describes the study of the existing works. Section 3 describes the research methodology. Section 4 addresses the state-of-the-art corpora utilized to carry out this classification problem. Section 5 presents an overview of the evaluation criteria. Section 6 discusses the numerous cutting-edge Machine learning approaches. Section 7 presents the author's observations regarding the proposed Research Questions. Finally, the work's conclusion is included in Section 8.

2. Literature Review

The RF model and various other ML techniques were proposed by Rao et al. in a novel way [6]. The overfitting issue and sparse or missing data can both be dealt with using the RF approach.

The logistic regression is reliable for finding able ways to find independent variables gathered for two groups. The features' recurrence, incompatibility, and the negative predictive consequences of outlier values are their examples of its limitations. However, the support vector machine approach can be used because it is better suited for various independent variables [7]. Large, noisy datasets are a constraint of this method, although it works well for nonlinear issues.

A survey of the main detection methods and taxonomy for phishing detection was presented by Vijayalakshmi et al. in 2020 [8]. According to an APWG data analysis, phishing attacks increased from 2017 to 2019. In the study, a taxonomy of automated phishing detection solutions was presented. Depending on the input parameters, the taxonomy divided all the solutions into three categories: web address-based methods, webpage content-based solutions, and hybrid approaches. Web address-based approaches were classified into list-based, heuristic rule-based, and learning-based approaches based on the techniques used in the solutions, and web content-based approaches were divided into rule-based and machine learning-based solutions.

Ozgur et al. [9] implemented web phishing classification by collecting their own data from the available resources. The Random Forest algorithm with solely NLP-based features performs the best with a 97.98% accuracy rate for phishing URL identification, according to experimental and comparative data from the implemented classification methods.

Jain et al. [10] proposed a methodology for performing the classification task in detecting phishing websites when compared to other machine learning approaches logistic regression classifier achieved more than 98.4% accuracy.

3. Methodology

Three datasets altogether were utilized to detect web phishing. The datasets were obtained from the machine learning repositories at UCI and Kaggle. These functions have a direct connection to website content. The datasets varied in size, which is crucial for assessing the precision and effectiveness of the Pre-processing is done one algorithm.

Each of processing is done to remove extraneous features and deal with missing values. Data is divided for training and testing. To create a Phishing Classifier model, we explored a variety of algorithms [36][37][38]. They include decision trees, Bernoulli's Naive Bayes, logistic regression, support vector machines, K-nearest neighbours, random forests, and ensembling techniques [41] like Gradient Boosting, Hist Gradient Boosting, AdaBoost, XGBoost, and LightGBM. Finally, test data are provided to validate the output of the algorithms. Several statistical indicators, including recall, accuracy, and precision, are used to assess performance. The adopted methodology is outlined in Fig. 2. this survey article are to investigate and comprehend the leading Machine Learning algorithms utilized for Phishing websites addition, as well as to react to a few research-related questions.

Q1. What recent datasets are available for this task?
Q2. Which evaluation techniques apply to this task?
Q3. What strategies may machine learning techniques be used to categorize websites?
Q4. What are the results?

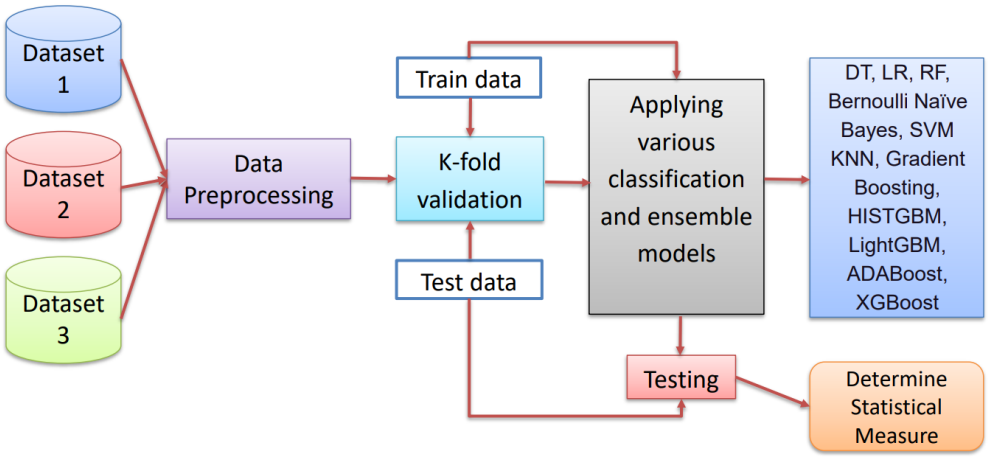

Figure 2. Flowchart of methodology

4. Datasets

Three datasets altogether were utilized to detect web phishing. The Kaggle and UCI machine learning repositories are where the datasets are pulled from [11][12]. The size of the datasets varies as well, which is crucial for assessing the precision and effectiveness of various algorithms.

Dataset - 1: The first dataset contains 2456 URLs with 28 attributes and is titled "Phishing Websites." The material does not specify a specific date of collection for the Phishing Websites Data Set. The dataset was given to the UCI Machine Learning Repository on March 26, 2015, as noted in the dataset description. The information in the dataset, however, might have been obtained from several sources over time prior to the donation date, including the MillerSmiles archive, the PhishTank archive, and Google's search operators. The precise dates of data collection for the dataset may vary based on the original sources and methods used to obtain the data.

Dataset - 2: Titled as Web page phishing detection dataset. The 48-feature dataset was created from 5000 authentic websites and 5000 fraudulent websites between January and May 2015 and May and June 2017. By utilizing the Selenium WebDriver browser automation framework, a better feature extraction method is used, which is more accurate and reliable than a parsing strategy based on regular expressions. It is appropriate for WEKA.

Dataset – 3: Titled as Phishing Dataset for machine learning. As a part of the dataset 87 characteristics were retrieved from the 11430 URLs. This dataset is designed to be a standard reference for machine learning-based phishing detection systems. It contains a total of 63 features, which are divided into three categories. Seven features are derived from communication with other services, while the remaining 56 features are based on the structure and syntax of URLs. The dataset is evenly balanced with an equal number of authentic and phishing URLs, making up 50% each.

Among these datasets, Dataset-1 is an unbalanced dataset. Remaining two are of balanced datasets.

Table 1. Summary of the datasets

	Total Number of records	Phishing Records	Non-Phishing /legitimate Records
Dataset-3(89)	11430	5715	5715
Dataset-2(50)	10000	5000	5000
Dataset-1(32)	11055	6157	4898

5. Evaluation Metrics

The effectiveness of machine learning algorithms for categorizing phishing websites can be assessed using a variety of evaluation approaches. Here are several regularly employed methods, including Precision, Recall, Accuracy and F1-Score [13][14].

A Confusion Matrix is a tabular representation of the counts of true positives, true negatives, false positives, and false negatives of the data. It is used to evaluate the effectiveness of a binary classifier.

Table 2. Confusion matrix

		Actual Value	
Precited Values		Phishing	Legitimate
	Phishing	**TP**	**FP**
	Legitimate	**FN**	**TN**

The most fundamental evaluation statistic, accuracy is determined by dividing the total number of predictions by the number of correct predictions generated by the model as shown in the equation (1). When the statistics are unbalanced, that is, when one class is far more numerous than the other, accuracy might be deceptive.

Precision is the ratio of actual positive results (phishing websites accurately identified as such) to all expected positive results (all websites identified as such) as in the equation (2).

Recall is the ratio of real positives—i.e., all the phishing websites in the dataset—to the overall number of positives as in the equation (3). When working with data that is unbalanced, it may be necessary to make a trade-off between the two measurements.

A more balanced way to evaluate the performance of the model than accuracy is to use the F1-Score, which is the harmonic mean of precision and recall as in the equation (4). When the dataset is unbalanced, it is frequently employed.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\_Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

6. Approaches

Following are some methods for classifying websites:

- Analysis of a URL's syntax and structure is required to establish the category of the URL. Phishing websites, for instance, may utilize URLs that match those of real websites [5].
- Link-based classification: To categorize a website, this technique looks at the links on it. The category could be determined by the machine learning algorithm by looking at the types of websites that the website links to [15][16].
- Machine learning-based analysis: This technique involves using a dataset of well-known legitimate and phishing websites to train a machine learning algorithm, which is then used to categorize new websites. The algorithm can learn to recognize patterns and characteristics that are typical of phishing websites, such as the use of specific URL structures or the presence of particular keywords [17][18].

Several machine learning techniques are used to categorize phishing websites [39][40], including the decision tree approach, which is straightforward and efficient and uses recursive partitioning of data to classify phishing websites. Multiple decision trees are combined to create Random forests, which uses predictions. Another well-liked technique for accurately identifying phishing websites is SVM. In binary classification tasks, such as classifying websites as legitimate or phishing, logistic regression is used. A probabilistic approach called Naive Bayes is used to categorize webpages. Based on the separation of the data points, KNN is also used to detect phishing websites. Gradient boosting is an ensemble method for increasing classification accuracy by combining the results of various decision trees [19][20][21].

Decision Tree: A popular Supervised machine learning method for categorizing web pages or URLs as legitimate or phishing is the decision tree algorithm. Variables like URL structure, URL length, port, and other variables are employed as predictors and it was trained on a labelled dataset. A decision tree represents all potential outcomes (also known as leaves) of a decision process (also known as branches). By dividing the dataset's best characteristics and criteria, the decision tree is constructed recursively. The Gini impurity is the default classification criterion used by the decision tree classifier.

The following are the steps to implement decision tree.

Step 1: *If every record in D_t is a member of class y_t, then t is a leaf node with the label y_t.*

Step 2: *To divide the records into more manageable groups, an attribute test condition is chosen if D_t contains records that belong to multiple classes. For each test condition outcome, a child node is formed, and records in D_t are assigned to the children based on the outcomes. Then, each child node receives a recursive application of the algorithm.*

Random Forest: A classification or regression problem's outcome can be predicted using the supervised machine learning method Random Forest. It is an ensemble method that creates several decision trees during training and only utilizes the most crucial attributes during prediction. Using features like n_estimators, criteria, and others, it predicts the input URL by iteratively going through each decision tree in the random forest. The final prediction is then decided by majority voting. To address the issue of overfitting in decision trees, the random forest technique was initially developed. This is accomplished by creating many decision trees, each one employing a different subset of the input data [22][23].

Naive Bayes: Is a supervised classification technique that categorizes unknown inputs using the Bayesian probability [24]. This model determines the conditional probability of a URL or web page being a legitimate or phishing site. Gaussian Nave Bayes, Bernoulli Nave Bayes, and more variations of Bayesian algorithms exist. When using Bernoulli Naive Bayes, which commonly uses binary features, the prior probabilities and likelihoods for each feature and each class are calculated. The final class label for the URL or webpage is projected to be the class with the highest posterior probability. The naïve Bayes classification is defined as
If X is a set of d attributes $X = \{x_1, x_2, \ldots, x_d\}$, The Naive Bayes classifier calculates the posterior probability for each class y to categories a test record.. The highest probability is the class that the test record belongs to.

$$P(Y|X) = \frac{P(y)\prod_{i=1}^{d} P(x_i|y)}{P(X)} \qquad (5)$$

Logistic Regression: A statistical technique for simulating the likelihood that an event will occur is logistic regression [24]. Given one or more independent factors, it is used to forecast the result of a categorical dependent variable. Any type of data can be used with this method. Utilizing a logistic function, logistic regression makes predictions about the likelihood of an occurrence.
The form of this function is:

$$P(Y = 1) = \frac{1}{1 + e^{-x}} \qquad (6)$$

Where x is the input variable and Y is the predicted outcome.

Where $P(Y = 0) = 1 - P(Y = 1)$. The value of x for which $P(Y = 0) = P(Y = 1)$is known as the point of inflection, or breakpoint.

Support Vector Machine: A supervised machine learning approach known as a support vector machine maximizes the shortest distance. SVM uses a linear kernel function to find an ideal hyperplane in an N-dimensional space that, depending on its feature space, can distinguish between authentic and phishing data the most effectively [25][26].

K-Nearest Neighbours: KNN Classifier is a supervised learning-based classification system that divides data into various categories [26]. This technique classifies each data point individually using its k nearest neighbours. The value of k in this issue is 3. To properly label each data point for the KNN classifier, we must first decide which category or class each piece of data should go under. These two pieces of information are fed into the KNN classifier, which uses the distance between each piece of data and its nearest k neighbours to determine which category it should fall under. The following are steps for implementing KNN.

- *Choose the number of nearest neighbours to consider, denoted as 'k'.*
- *Prepare a set of labeled training examples, denoted as D.*
- *For each test example z:*
  - *a. Compute the distance between z and every example (x, y) in D.*
  - *b. Select the k closest training examples to z.*
  - *c. Determine the most frequent class label from the k nearest neighbours, and assign it as the predicted label for z.*
- *Repeat step 3 for all test examples.*
- *End.*

The algorithm computes the distance between each test example $z = (x_1, y_1)$ and all training examples $(x, y) \in D$ to determine its nearest neighbour list D_z.

Gradient Boosting: When there are numerous features with significant levels of association, this model is frequently used [27]. The classifier gains the ability to match the features of websites to their labels (such as legitimate or phishing). Multiple weak learners are combined using gradient boosting to produce a powerful model. When the target level of accuracy is attained, the algorithm stops adding weak classifiers and starts over with an empty model [28].

Extreme Gradient Boosting Classifier: An ensemble-based classification system called XGBoost Classifier can be applied to any machine learning issue involving sizable datasets. It builds powerful classifiers using a boosting method. By analyzing the elements of the website, XG enhances and trains the data based on the learning rate and maximum depth, and then assigns a score that indicates the likelihood that the website is a phishing website. The primary flaw in XG boost is its inability to handle categorical characteristics.

Histogram Gradient Boosting Classifier: For large datasets (sample >=10000), Histogram Gradient Boosting outperforms the Gradient Boosting Classifier, which combines many weak learning models. Max_bins, Max_depth, and other terms are employed in this classifier, and default values are considered. The tree grower learns at each split whether to choose the left or right child (i.e., phishing or legitimate as the ultimate split) based on the prospective gain. Those samples are mapped to the child with

the most samples if there are no missing values discovered during training.

Light Gradient Boosting: The XGBoost method, which also manages unbalanced datasets, is comparable. It was created by Microsoft, is quicker, and uses less memory. As light GBM develops vertically (leaf-wise), more loss is reduced [29].

Adaptive Boosting: AdaBoost Classifier is a machine learning technique with an ensemble approach to categorize fresh data points. Data is trained using n_estimators and the learning rate [30]. Ada boost produces stumps, a tree with only two leaves. Stumps' principal function is to eliminate errors; however, they are not given equal weight in the final decision tree. All the data points are first given equal weights. When classification is done incorrectly, weights are increased. By sequential training on the training data and subsequently testing on the test data, it iteratively creates a classifier.

Various machine learning algorithms along with the optimization techniques can be used to perform web phishing classification. As the dimensions of the dataset also play a key role. If the dataset is of multidimensional feature, it may lead to the overfitting condition [31]. So, the dimensions can be reduced to improve the model's accuracy.

7. Results

From the table 1, XG Boost has the highest accuracy in dataset 1 with a score of 97.13%. This algorithm ran in 1.51 seconds. Logistic Regression, which has an accuracy rate of 91.73%, is the least accurate algorithm.

Table 1. Dataset1 Performance

ML Model	Accuracy	Recall	Precision	F1_Score	Execution Time (Sec)
Decision Tree	95.62	0.9655	0.9579	0.9617	0.25339
Random Forest	96.65	0.9782	0.9635	0.9708	0.08086
Bernoulli-naive_bayes	92.46	0.9444	0.9248	0.9345	0.01561
Logistic Regression	91.73	0.9221	0.9320	0.9270	0.04559
Support_Vector_Machine	92.16	0.9343	0.9284	0.9313	0.92638
KNeighbors Classifier	94.45	0.9560	0.9470	0.9515	0.50498
Gradient Boosting	94.78	0.9565	0.9520	0.9543	0.73755
HistGradient Boosting	96.65	0.9761	0.9654	0.9707	1.55873
LightGradient Boosting	96.59	0.9756	0.9649	0.9702	1.48017
AdaBoost	93.78	0.9491	0.9421	0.9456	0.53470
XGBoost	97.13	0.9846	0.9657	0.9750	1.51306

Table 2. Dataset 2 Performance

ML Model	Accuracy	Recall	Precision	F1_Score	Execution Time (Sec)
Decision Tree	85.63	0.7430	0.9694	0.8412	0.17286
Random Forest	97.26	0.9596	0.9866	0.9729	0.07845
Bernoulli-naive_bayes	88.46	0.8874	0.8874	0.8874	0.01601
Logistic Regression	94.30	0.9473	0.9417	0.9445	0.10953
Support_Vector_Machine	94.53	0.9473	0.9460	0.9466	0.72356
KNeighbors Classifier	94.96	0.9479	0.9535	0.9507	0.39253
Gradient Boosting	95.23	0.9212	0.9847	0.9519	1.13267
HistGradient Boosting	90.93	0.8301	0.9914	0.9036	1.77236
LightGradient Boosting	**97.83**	0.9681	0.9893	0.9786	1.08554
AdaBoost	95.70	0.9342	0.9808	0.9570	0.77001
XGBoost	95.66	0.9258	0.9888	0.9563	0.81923

Table 3. Dataset 3 Performance

ML Model	Accuracy	Recall	Precision	F1_Score	Execution Time (Sec)
Decision Tree	90.87	0.8803	0.9314	0.9051	0.23539
Random Forest	94.69	0.9204	0.9707	0.9449	0.12570
Bernoulli-naive Bayes	89.06	0.8821	0.8953	0.8886	0.01451
Logistic Regression	94.66	0.9457	0.9463	0.9460	0.18761
Support Vector Machine	94.86	0.9457	0.9502	0.9480	1.68424
K-Neighbors Classifier	94.51	0.9322	0.9558	0.9439	0.48715
Gradient Boosting	95.88	0.9505	0.9658	0.9581	2.87812
HistGradient Boosting	96.90	0.9593	0.9777	0.9684	2.53556
LightGradient Boosting	96.90	0.9599	0.9772	0.9684	1.41398
AdaBoost	94.66	0.9357	0.9554	0.9455	1.63595
XGBoost	**97.05**	0.9675	0.9727	0.9701	1.36775

From the table 2. Light Gradient Boosting and Random Forest performed best in this dataset, scoring 97.83% and 97.26%, respectively. The algorithm that required the least amount of processing time is Random Forest. It completed in 0.078 seconds as opposed to the light gradient boost's 1.08 seconds and high F1 score of 0.978%. Decision Tree is the least accurate algorithm, with an accuracy rate of 85.63%.

The only algorithm with accuracy greater than 97.05% is XG Boost. With 89.06% accuracy, Bernoulli naive bayes is the algorithm that has a lower percentage of accuracy when compared to other algorithms and takes very little time to perform.

Among all the datasets the performance of the UCI Phishing dataset is yielding better results compare to other datasets as shown in the figure. Even advanced topics like deep learning [32][33], transfer learning can be used to classify the web phishing websites [34].

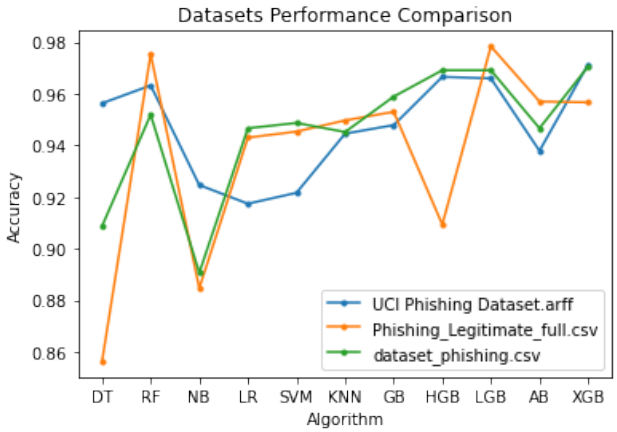

Figure 1. Performance of the datasets on different machine learning techniques

8. Conclusion

This research paper describes how machine learning algorithms effectively detect and predict web phishing attacks. The analysis of different machine learning techniques can accurately classify phishing websites based on the various features such as path, URL, domain name, sub-domain name, and directory. For instance, decision tree models can effectively identify the relevant features for classification, while random forest can improve the accuracy and robustness of the classification models, SVM can easily handle highly dimensional feature spaces. Boosting methods like gradient boosting, XG boost, and Ada boost have shown that these algorithms can accurately classify into phishing or legitimate. Boosting algorithms are highly effective in improving the performance of weaker machine-learning algorithms. By iteratively reweighting the training examples, boosting algorithms can boost the model's accuracy by giving more weight to different examples. Other techniques like deep learning and neural networks [42][43] can also be used in further works. But this paper mainly focuses on the machine learning and boosting algorithms that can be done easily with less complexity. Overall, machine learning algorithms can significantly enhance the security of web users by providing phishing detection. As cybercriminals continue to develop more sophisticated phishing attacks, the use of machine learning algorithms will become increasingly important in ensuring the safety and security of online users.

References

[1] Odeh A, Keshta I, Abdelfattah E. Machine Learning Techniques for detecting website phishing: A review for promises and challenges. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE; 2021.

[2] Chiew KL, Tan CL, Wong K, Yong KSC, Tiong WK. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Inf Sci (Ny) [Internet]. 2019;484:153–66. Available from: http://dx.doi.org/10.1016/j.ins.2019.01.064

[3] H. Bleau, "Global fraud and cybercrime forecast," ed: Retrieved from RSA: https://www. rsa. com/en-us/resources/2017-global-fraud/… 2017

[4] Sirisha A, Nihitha V, Deepika B. Phishing URL detection using machine learning techniques. In: Lecture Notes in Electrical Engineering. Singapore: Springer Nature Singapore; 2021. p. 1067–80
[5] Feroz MN, Mengel S. Phishing URL detection using URL ranking. In: 2015 IEEE International Congress on Big Data. IEEE; 2015.
[6] Rao RS, Pais AR. Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput Appl [Internet]. 2019;31(8):3851–73. Available from: http://dx.doi.org/10.1007/s00521-017-3305-0
[7] Sahingoz OK, Buber E, Demir O, Diri B. Machine learning based phishing de-tection from URLs. Expert Systems with Applications. 2019;117:345–57.
[8] Vijayalakshmi M, Mercy Shalinie S, Yang MH, Meenakshi R. Web phishing detection techniques: a survey on the state-of-the-art, taxonomy, and future directions. IET Netw [Internet]. 2020;9(5):235–46. Available from: http://dx.doi.org/10.1049/iet-net.2020.0078
[9] Sahingoz O, Koray E, Buber O, Demir B. Machine learning based phishing detection from URLs. Expert Systems with Applications. 2019;117:345–57.
[10] Jain AK, Gupta BB. A machine learning-based approach for phishing detection using hyperlinks information. J Ambient Intell Humaniz Comput [Internet]. 2019;10(5):2015–28. Available from: http://dx.doi.org/10.1007/s12652-018-0798-z
[11] Vrbančič G, Fister I Jr, Podgorelec V. Datasets for phishing websites detection. Data Brief [Internet]. 2020;33(106438):106438. Available from: http://dx.doi.org/10.1016/j.dib.2020.106438
[12] Karabatak M, Mustafa T. Performance comparison of classifiers on reduced phishing website dataset. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS). IEEE; 2018. p. 1–5.
[13] Odeh AJ, Keshta I, Abdelfattah E. Efficient detection of phishing websites using multilayer perceptron. Int J Interact Mob Technol [Internet]. 2020;14(11):22. Available from: http://dx.doi.org/10.3991/ijim.v14i11.13903
[14] Hossin M, Nasir Sulaiman M. International journal of data mining & knowledge management process. 2015;5.
[15] Becchetti L, Castillo C, Donato D, Leonardi S, Ba-Eza-Yates RA. Link-based characterization and detection of web spam. In: AIRWeb. 2006. p. 1–8.
[16] Roul RK, Asthana SR, Shah M, Parikh D. Detecting spam web pages using content and link-based techniques. Sadhana [Internet]. 2016;41(2):193–202. Available from: http://dx.doi.org/10.1007/s12046-015-0460-9
[17] Shahrivari V, Darabi MM, Izadi M. Phishing detection using machine learning techniques. 2020.
[18] Zamir A. Phishing website detection using various machine learning algorithms. In: The Electronic Library. 2020.
[19] Singh J, Singh J. A survey on machine learning-based malware detection in executable files. Journal of Systems Architecture. 2020;
[20] Jyothi UP, Dabbiru M, Bonthu S, Dayal A, Kandula NR. Comparative analysis of classification methods to predict diabetes mellitus on noisy data. In: Lecture Notes in Electrical Engineering. Singapore: Springer Nature Singapore; 2023. p. 301–13.
[21] Silpa, Rao DVVRM. Enriched big data pre-processing model with machine learning approach to investigate web user usage behavior. Indian J Comput Sci Eng [Internet]. 2021;12(5):1248–56. Available from: http://dx.doi.org/10.21817/indjcse/2021/v12i5/211205050
[22] Akinyelu AA, Adewumi AO. Classification of phishing emails using random forest machine learning technique. J Appl Math [Internet]. 2014;2014:1–6. Available from: http://dx.doi.org/10.1155/2014/425731
[23] Subasi A, Molah E, Almkallawi F, Chaudhery TJ. Intelligent phishing website detection using random forest classifier. In: 2017 International Conference on Electrical and computing technologies and Applications (ICECTA). IEEE; 2017. p. 1–5.
[24] Othman N, Fadzilah WISW. Youtube spam detection framework using naïve Bayes and logistic regression. Indonesian Journal of Electrical Engineering and Computer Science. 2019;14(3):1508–17.
[25] Zouina M, Outtaj B. A novel lightweight URL phishing detection system using SVM and similarity index. Hum-centric Comput Inf Sci [Internet]. 2017;7(1). Available from: http://dx.doi.org/10.1186/s13673-017-0098-1
[26] Altaher A. Phishing websites classification using hybrid SVM and KNN approach. International Journal of Advanced Computer Science and Applications. 2017;8(6).
[27] Stobbs J, Issac B, Jacob SM. Phishing web page detection using optimized machine learning. In: 2020 IEEE 19th International Conference on Trust, Security, and Privacy in Computing and Communications (TrustCom). IEEE; 2020.
[28] Pavan R, Nara M, Gopinath S, Patil N. Bayesian optimization and gradient boosting to detect phishing websites. In: 2021 55th Annual Conference on Information Sciences and Systems (CISS). IEEE; 2021.
[29] Oram E, Dash PB, Naik B, Nayak J, Vimal S, Nataraj SK. Light gradient boosting machine-based phishing webpage detection model using phisher website features of mimic URLs. Pattern Recognit Lett [Internet]. 2021;152:100–6. Available from: http://dx.doi.org/10.1016/j.patrec.2021.09.018
[30] Subasi A, Kremic E. Comparison of AdaBoost with MultiBoosting for phishing website detection. Procedia Comput Sci [Internet]. 2020;168:272–8. Available from: http://dx.doi.org/10.1016/j.procs.2020.02.251
[31] Yang P, Zhao G, Zeng P. Phishing website detection based on multidimensional features driven by deep learning. IEEE Access [Internet]. 2019;7:15196–209. Available from: http://dx.doi.org/10.1109/access.2019.2892066
[32] Feng F, Zhou Q, Shen Z, Yang X, Han L, Wang J. The application of a novel neural network in the detection of phishing websites. J Ambient Intell Humaniz Comput [Internet]. 2018; Available from: http://dx.doi.org/10.1007/s12652-018-0786-3
[33] Pan Y, Sun F, Teng Z, White J, Schmidt DC, Staples J, et al. Detecting web attacks with end-to-end deep learning. J Internet Serv Appl [Internet]. 2019;10(1).

Available from: http://dx.doi.org/10.1186/s13174-019-0115-x
[34] Sridevi S. Improving the performance of automatic short answer grading using transfer learning and augmentation. Artificial Intelligence. 2023;123.
[35] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab, "Knowledge-Driven Cybersecurity Intelligence: Software Vulnerability Coexploitation Behavior Discovery," in IEEE Transactions on Industrial Informatics, vol. 19, no. 4, pp. 5593-5601, April 2023, doi: 10.1109/TII.2022.3192027.
[36] Basnet, Ram, Srinivas Mukkamala, and Andrew H. Sung. "Detection of phishing attacks: A machine learning approach." Soft computing applications in industry (2008): 373-383.
[37] Tang, Lizhen, and Qusay H. Mahmoud. "A survey of machine learning-based solutions for phishing website detection." Machine Learning and Knowledge Extraction 3.3 (2021): 672-694.
[38] Dutta, Ashit Kumar. "Detecting phishing websites using machine learning technique." PloS one 16.10 (2021): e0258361.
[39] Salloum, Said, et al. "Phishing email detection using natural language processing techniques: a literature survey." Procedia Computer Science 189 (2021): 19-28.
[40] Safi, Asadullah, and Satwinder Singh. "A Systematic Literature Review on Phishing Website Detection Techniques." Journal of King Saud University-Computer and Information Sciences (2023).
[41] Ullah, Zahid, and Mona Jamjoom. "A smart secured framework for detecting and averting online recruitment fraud using ensemble machine learning techniques." PeerJ Computer Science 9 (2023): e1234.
[42] Rasool, Raihan Ur, et al. "CyberPulse++: A machine learning-based security framework for detecting link flooding attacks in software defined networks." International Journal of Intelligent Systems 36.8 (2021): 3852-3879.
[43] Guo, Yanhui, Zelal Mustafaoglu, and Deepika Koundal. "Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms." Journal of Computational and Cognitive Engineering 2.1 (2023): 5-9.
[44] Barry Adams (2022) Everything Publishers Need to Know About URLs. In: seoforgooglenews. https://www.seoforgooglenews.com/p/everything-urls-news-publishers/. Accessed 9th Feb 2022.
[45] You, M., Yin, J., Wang, H. et al. A knowledge graph empowered online learning framework for access control decision-making. World Wide Web 26, 827–848 (2023). https://doi.org/10.1007/s11280-022-01076-5
[46] Yin, J., Tang, M., Cao, J. et al. Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning. World Wide Web 25, 401–423 (2022). https://doi.org/10.1007/s11280-021-00909-z
[47] You, M., Yin, J., Wang, H., Cao, J., Miao, Y. (2021). A Minority Class Boosted Framework for Adaptive Access Control Decision-Making. In: Zhang, W., Zou, L., Maamar, Z., Chen, L. (eds) Web Information Systems Engineering – WISE 2021. WISE 2021. Lecture Notes in Computer Science(), vol 13080. Springer, Cham. https://doi.org/10.1007/978-3-030-90888-1_12
[48] Yin, J., Tang, M., Cao, J., Wang, H., You, M., Lin, Y. (2020). Adaptive Online Learning for Vulnerability Exploitation Time Prediction. In: Huang, Z., Beek, W., Wang, H., Zhou, R., Zhang, Y. (eds) Web Information Systems Engineering – WISE 2020. WISE 2020. Lecture Notes in Computer Science(), vol 12343. Springer, Cham. https://doi.org/10.1007/978-3-030-62008-0_18